Do It Yourself: Standalone Pseudo-labeling for Domain Adaptation in Deep Computer Vision

Tsirmpas Dimitris

11th May 2024

1 Introduction

2 Related Works

3 Pseudo-labeling for Classification Enhancement

3.1 Model-agnostic Pseudo-labeling algorithm

We use a modified version of the iCAN algorithm (?). The original iCAN algorithm was designed around a model created with the explicit purpose of learning domain-invariant features between the source and target datasets (CAN model). When the unmodified algorithm uses a generic deep model however a number of issues arise. Our approach seeks to remedy the most major such issues.

First of all, in the original paper, the authors propose adding the loss of the fully supervised dataset $\mathcal{L}_{source}$, the loss of the target dataset $\mathcal{L}_{tar}$ and the loss of their model's domain-agnostic penalty $\mathcal{L}_{CAN}$ for each mini-batch as $\mathcal{L} = \mathcal{L}_{source} + \mathcal{L}_{tar} + \mathcal{L}_{CAN}$.

The modified procedure can be found in Algorithm 1, where D_{source} is the source dataset containing the training instances, D_{target} is the equivalent target dataset and $\%$ is the modulo operator.

Algorithm 1 Modified general incremental learning algorithm

Train model on dataset D_{source}
D_{pseudo} = $\{\}$
for epoch **do**
    if $epoch\%N_{period} = 0$ **then**
        Calculate $accuracy$ on the validation source dataset
        $\mathcal{T} = adaptive\_threshold(accuracy)$
        Select all samples $d \in D_{target}$ where $model(d) > \mathcal{T}$
        Add all selected samples to D_{pseudo}
        Remove all selected samples from D_{target}
    end if
    Select random samples from D_{source} and add to D_{rand_source} such as $|D_{rand\_source}| = |D_{pseudo}|$
    Train epoch on D_{rand_source}
    Train epoch on D_{pseudo}
end for